Auragen Systems Corp.


# AURAGEN FAULT TOLERANCE

The AURAGEN™ System eliminates the problems and costs previously associated with fault tolerant computing. In the past, fault tolerant computers were either significantly more expensive or more difficult to use. With its unique architecture and fault tolerant operating system, AURAGEN provides not only the most productive, user-friendly fault tolerant computer, but also a viable, cost-effective alternative to conventional computer systems.

All fault tolerant systems require hardware duplication so that if one part fails another is immediately available to take over and continue its work. Duplication is also required for programs. Every program in the system has two copies. The primary does the actual processing. The backup can continue the work of the primary if it should fail.

Fault tolerant systems differ in the cost of providing duplicate hardware and software. In some systems the additional hardware yields no additional performance. Other systems require special knowledge to write fault tolerant software.

The earliest approach to fault tolerance utilizes duplexed processors. In effect, there are two systems working on the same problem; primary and backup programs execute simultaneously on separate processors. While such systems can be programmed as easily as conventional computers, the major disadvantage of even modern implementations of this old design concept is that they can only provide half the price/performance of a non-duplexed processor.

More recent approaches to fault tolerance involve maintaining a non-executing backup on a separate processor. In fault tolerant systems of this kind, the second processor may do productive work because the backup is inactive. It is necessary, however, to keep the backup informed about the state of the primary to insure that the backup can take over in case the primary fails.

The most common method of keeping the backup informed is checkpointing. The primary program periodically sends the backup a copy of its stack and data. The disadvantages of checkpointing are:

- Primary programs are slowed down by sending checkpoints.
- Backups are not fully available to do productive work because they must receive and process checkpoints.
- Special knowledge is required to write checkpointing programs.

## The AURAGEN Advantage

Recent advances in hardware and software technology have enabled AURAGEN to build a system that is more efficient in maintaining backup processes than any other fault tolerant system.

In the AURAGEN System, backups are non-executing. A backup is kept informed about the state of the primary by receiving all messages (i.e., program input) sent to the primary. If the primary fails, the backup can recreate and continue the work of the primary. Since the backup processor is only receiving messages and saving them for use in case of a failure, the backup is fully available to do productive work during normal operation. The primary is not slowed down by sending frequent checkpointing messages.

The monitoring and queuing of messages for fault tolerance is handled automatically by the Message System which is fully integrated with AUROS,™ the AURAGEN operating system. Consequently, the use of messages to support fault tolerance is transparent to both end-users and programmers. No special programmer knowledge is required to write and run AURAGEN fault tolerant programs. Moreover, since AUROS was derived from and is compatible with UNIX,™ programs written for standard UNIX will run in fault tolerant mode on the AURAGEN System without modification.

In an AURAGEN cluster, the Message System runs primarily in the Executive Processor module, while user programs run in the Work Processor module. Because of this separation, the Work Processor incurs very little overhead for fault tolerance, and user programs can make optimal use of processing resources. The Message

System also makes more memory available for executing user programs by ensuring that backup processes are paged out of main memory during normal processing.

## Message System Operations

The Message System assures that a single message sent from one primary process to another is simultaneously received at three destinations:

- The requested destination process.
- The backup of the destination process.
- The backup of the sending process.

No processing occurs unless all three messages are delivered. Upon receipt of the message, different actions are taken at each destination:

- The message is queued up for processing by the primary process.
- The message is queued up and saved for use by the backup process in case a recovery is necessary.
- The backup of the sending process records that a message has been sent, and then discards the message.

Only a minimal amount of system resources is used by the Message System to maintain the message queues needed for fault tolerant computing.

During normal processing, backup processes are kept in a non-executing state that is *almost identical* to the state of their corresponding executing processes. Should a failure occur (a rare event), a backup process brings itself up to the exact state of its primary by using information made available through the Message System. In particular, the backup catches up by reprocessing the primary's input messages which are accessible from its message queue. Complete recomputation upon failure is avoided by periodically synchronizing backup and primary processes.

In addition to providing backups with message information needed to take over in case of failure and controlling periodic synchronization, the Message System assures that upon recovery, backups interact correctly with the rest of the system. Specifically, the Message System makes sure that backup processes read the available messages in *exactly* the same order as the primary. The Message System also ensures that when a backup process updates itself it does not resend any messages already sent by the primary.

## Synchronizing Primary and Backup Processes

Backup processes are periodically updated to match the state of their corresponding primaries. This synchronization entails invoking the normal page fault mechanism to update those pages changed in real memory but not yet updated in virtual memory. Synchronization also frees memory by discarding queued backup messages already read by primaries. Execution of primary programs is not slowed because programs need not wait for synchronization to be complete.

During normal operation one of two events trigger synchronization:

- The primary process has read a system defined number of messages, or
- The primary process has executed a system defined number of instructions since the last synchronization.

The performance of the AURAGEN System can be optimized by changing these synchronization parameters.

## Fault Tolerant Recovery Procedures

When a failure occurs, the Message System notifies all clusters holding needed backup processes that fault tolerant recovery procedures must be executed. A backup begins executing in the state that the failed primary had achieved *at the time of last synchronization*. The Message System, via its management of both primary and backup message queues, guarantees that:

- The right messages will be available to backups.
- The messages will be executed in the right order.
- The backup will not resend any messages that were generated by the primary between last synchronization and failure.

The entire fault tolerant recovery procedure occurs in a few seconds *without end-user knowledge, without manual intervention, and with limited system overhead.*

The AURAGEN approach to fault tolerance produces a system that offers more benefits than any other fault tolerant system. The AURAGEN Message System makes efficient use of the additional computing power gained from duplicate hardware. It provides fully transparent, automatic fault tolerant processing. By enhancing UNIX to support fault tolerance, AURAGEN can provide reliable performance, reduced training costs, and continued compatibility with future versions of UNIX. All these features make AURAGEN not only the most productive and user-friendly fault tolerant computer available, but also a viable, cost-effective alternative to conventional computer systems.



Direct Sales and Service in U.S.A. are available from:

Auragen Systems Corp.
Two Executive Drive, Fort Lee, N.J. 07024 (201) 461-3400

200.FT.483